UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/847,063	04/30/2001	Ming Zhou	GEI-001US 29083	4555

21718 7590 07/20/2006

LEE & HAYES PLLC
SUITE 500
421 W RIVERSIDE
SPOKANE, WA 99201

EXAMINER	
RUTTEN, JAMES D	

ART UNIT	PAPER NUMBER
2192	

DATE MAILED: 07/20/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/847,063	ZHOU ET AL.	
	Examiner	Art Unit	
	J. Derek Rutten	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *28 April 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *20-24,26,32-35,40,49,52,55 and 57-72* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *20-24,26,32-35,40,49,52,55 and 57-72* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *30 April 2001* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☒ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to Applicant's submission filed 4/28/2006, responding to the 12/2/2005 Office action which detailed the rejection of claims 1-4, 7-11, 13-15, 18-24, 26, 32-35, 39-43, 49, 51, 52, and 55. Claims 20, 21, 23, 24, 26, 32, 33, 35, 40, 49, 52, and 55 have been amended, claims 1-19, 25, 27-31, 36-39, 41-48, 50, 51, 53, 54, and 56 have been canceled, and new claims 57-72 have been added. Claims 20-24, 26, 32-35, 40, 49, 52, 55, and 57-72 remain pending in the application and have been fully considered by the examiner.

2. Applicant has primarily argued (see pages 16-21 filed 4/28/2006) that the claims are not obvious over the Hinks, Sun, Lyapustina references. This arguments is not persuasive, as will be addressed under the *Response to Arguments* section below.

3. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Response to Arguments

4. Applicant argues on page 12, filed 4/28/2006, that the reference to 37 CFR 1.56(a) satisfies the requirements of 37 CFR 1.56 (b) and (c) as well. This argument is persuasive. However, Applicant has not provided any argument that reference to 37 CFR 1.56(a) is sufficient for 37 CFR 1.56 (d) and (e) as well. Applicant further suggests that the deviations in the declaration are minor and could be waived according to MPEP § 602.03. However, MPEP § 602.03 states that *minor* deficiencies in the body of the oath or declaration where the deficiencies are self-evidently cured in the rest of the oath or declaration can be waived when an application is otherwise ready for issue. Applicant has not provided any reasons why the omission of 37 CFR 1.56 (d) and (e) should be considered minor. Further, the application is not otherwise ready for issue. Therefore, these arguments are not persuasive, and the objection is maintained.

5. Applicant essentially suggests in paragraph 2 on page 17 that the Sun reference does not "remove the locale-sensitive content as claimed". This argument was addressed in the previous Office Action (paragraph 4, pages 2 and 3, mailed 12/2/2005) and was persuasive. Applicant further suggests that since the locale-sensitive content in Sun is not removed, it could not further be *substituted* with a function call. It is noted that this statement reflects a presentational view of source code, and that Sun teaches the semantic substitution of locale-sensitive content (See Sun, page 100). While these arguments are otherwise convincing, Sun is not relied upon to teach these limitations. The Lyapustina reference teaches that one sequence of characters can be substituted for another sequence of characters (See Lyapustina column 4 lines 18-22).

6. Applicant essentially argues (see paragraph 2 page 18) that Lyapustina's unique macro string does not operate as a "function call", and that it is used at compile time, and not at runtime in the manner of claim 20. However, Lyapustina is not relied upon for teaching these limitations. Sun teaches the use of a function call in place of locale-sensitive content (Sun page 100). Therefore, this argument is moot.

7. From line 20 on page 18 through line 21 on page 21 of Applicant's response, Applicant argues that there is no motivation to combine the references. In response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992).

In this case, Applicant essentially argues that the Hinks reference teaches away from the Sun and Lyapustina references since they involve merely "extracting 'translatable strings' ..., translating them ..., and reinserting them ..." which Hinks identifies as having "distinct disadvantages" (see Hinks column 2 lines 30-44 as cited by Applicant). However, Hinks provides this discussion in terms of tools that focus only on "character-based information" (see Hinks column 2 lines 45-57 as cited by Applicant). Further inspection of the Sun reference reveals a teaching of not only the extraction, translation, and reinsertion of translatable strings, but also the sizing and display of such translated strings (see Sun page 100: "Size and Position of Elements"). Rather than teaching away, this provides further motivation to combine, since Hinks

discloses that displaying and printing is needed in addition to translation (see column 1 lines 30-35 as cited by Applicant).

Applicant further essentially argues that since Sun operates at runtime, and Lyapustina operates at compile time, that they do not complement each other and are mutually exclusive (see page 20 of Applicant's response). However, both references are concerned with the coded representation of strings in source code which exists independently from a particular compile-time or run-time. Further, Applicant's argument does not diminish the teaching of Lyapustina, that the hard coding of strings is disadvantageous and inflexible (see Lyapustina column 1 lines 64-65).

Therefore, for the above reasons, Applicant's argument that there is no motivation to combine, is not persuasive.

Claim Objections

8. Claims 58 and 70 are objected to because of the following informalities: They contain the word "sever", which appears to be a typo for --server--. Appropriate correction is required.

Drawings

9. The drawings are objected to under 37 CFR 1.83(a). The drawings must show every feature of the invention specified in the claims. Therefore, the "request contains an identity of a desired locale" (claims 57, 61, 65, and 69) must be shown or the feature canceled from the claims. No new matter should be entered.

Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Claim Rejections - 35 USC § 103

10. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

11. Claims 20-24, 26, 32-35, 49, 52, and 55 are rejected under 35 U.S.C. 103(a) as being unpatentable over prior art of record U.S. Patent 5,678,039 to Hinks et al. (hereinafter referred to as "Hinks"), in view of prior art of record "OpenWindows Developer's Guide: Xview Code

Generator Programmer's Guide" by Sun Microsystems (hereinafter referred to as "Sun") further in view of prior art of record U.S. Patent 6,802,059 to Lyapustina et al. (hereinafter "Lyapustina").

As per claim 20, Hinks discloses a method (FIG. 12A). Hinks further discloses:

compiling a computer-servable document written for a particular locale to extract ... characters associated with any original locale-sensitive content, the compiling producing a compiled document with locale-independent elements See column 3 lines 1-4, and 7-9:

> The Export/Import module itself includes a parsing engine to **extract** strings and translatable information from application programs.
> ...
> First, the sources are **parsed** by the Export/Import module to a translatable format.

Also FIG. 3 element 377 and column 8 lines 30-34:

> Alternatively, a Resource Compiler 365, again such as Borland's Resource Workshop®, may be employed to **re-compile** the Translated Resource Files 360 and bind those compiled resources back into the target program, now shown as Translated Program 377.

storing the original locale-sensitive content See column 3 lines 15-22:

> From there, Export/Import (EXPIMP) module parses the resource file into a Translation Table, which is typically **stored** as a database table. The Translation Table encapsulates all the information that is known or can be derived from the various resources and stores them in a format which may be utilized by various **editors**.

...

at runtime, retrieving the compiled document and populating the compiled document with a desired version of the original locale-sensitive content ... wherein the populating comprises executing in order to *obtain the desired version of the associated original locale-sensitive content and to insert the desired version of the associated*

original local-sensitive content back into the compiled document. See column 3 lines 50-55:

> Resources for products to be translated are stored in an external resource file in a standard format. Thus, the translation process need not interfere with the executable binary (i.e., main program) of a product. In this fashion, changing a product from one locale to another can be reduced to the simple process of swapping out resource files.

Hinks does not expressly disclose *removing characters* and *substituting a function call in place of associated locale-sensitive content in the compiled document; ... wherein the populating comprises executing the function call in the compiled document.*

However, Sun teaches *substituting a function call in place of associated ... original locale-sensitive content in the compiled document* See page 99 under the header "xgettext and msgfmt Utilities":

> Once Devguide or a developer has inserted `gettext()` function calls around all user-visible text in an application, `xgettext` can be run on the source files to produce the portable object files.

and the populating comprises executing the function call in the compiled document (page 98 under the header "gettext() and dgettext() Routines":

> Two similar routines are available to a developer for retrieving translated text. One, `gettext()`, assumes a text domain has already been specified by a call to a function called `textdomain()`. For example:
>
> ```
> Textdomain ("domain_name");
> .
> .
> gettext(Message_1\n");
> gettext("Message_2\n");
> ```

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Sun's function call substitution and population in Hinks' localization system. One of ordinary skill would have been motivated to include nothing specific to one's language and culture in software development and to provide features that facilitate translation of text into other languages (See Sun page 95).

Hinks and Sun do not expressly teach *removing* locale-sensitive content. However, in an analogous environment, Lyapustina teaches that locale-sensitive content can be removed and substituted with a unique identifier string. See column 4 lines 18-22:

> Upon identifying each string, the conversion mechanism generates a unique macro string as a substitute for the original string. The conversion mechanism then substitutes the unique macro string for the identified string in the source code of the computer program.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Lyapustina's unique macro string with Sun's function call. One of ordinary skill would have been motivated to avoid the use of hard coded strings (See Lyapustina column 1 line 64 – column 2 line 7).

As per claim 21 the above rejection of claim 20 is incorporated. Further, Hinks discloses *wherein the original locale-sensitive content comprises natural language text* (column 3 lines 28-31).

As per claim 22 the above rejection of claim 20 is incorporated. Further, Hinks discloses *wherein the locale-independent elements comprise source code and formatting data* (column 3 lines 31-34).

As per claim 23, the above rejection of claim 20 is incorporated. Further, Hinks discloses *storing the original locale-sensitive content in a structured text file* (column 3 lines 9-11. Resource files are structured text files.).

As per claim 24, the above rejection of claim 20 is incorporated. Further, Hinks discloses *storing the locale sensitive content in a database file* (column 3 lines 13-16).

As per claim 26, the above rejection of claim 20 is incorporated. Hinks further discloses:

storing one or more translated versions of the original locale-sensitive content corresponding to one or more respective other locales (column 3 lines 23-24).

As per claim 41, Hinks discloses:

A compiler system (FIG. 3) *comprising:*

a grammar containing rules for structuring source code (column 3 lines 7-9 as cited in the above rejection of claim 1 describes a parser which inherently uses a source code grammar, otherwise it would be unable to match tokens of the source code.);

a call library to store function calls (column 5 lines 26-29 describe use of the Microsoft Windows environment, which inherently contains a call library);

content analyzer to analyze source code in a document written for a particular locale and to utilize the grammar to determine whether the source code contains locale-sensitive content that is specific to the particular locale (column 3 lines 7-9 as cited in the above rejection of claim 1),

Hinks does not expressly disclose *the content analyzer being configured to the locale-sensitive content in the source code.*

However, in an analogous environment, Sun teaches *the content analyzer being configured to <manipulate> the locale-sensitive content in the source code with associated references to the replaced locale-sensitive content* (page 99 under the header "xgettext and msgfmt Utilities", and page 98 under the header "gettext() and dgettext() Routines" as referenced in the above rejection of claim 6).

All further limitations have been addressed in the above rejection of claim 1.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Sun's reference replacement in Hinks' localization system. One of ordinary skill would have been motivated to include nothing specific to one's language and culture in software development and to provide features that facilitate translation of text into other languages.

As per claim 42, the above rejection of claim 41 is incorporated. Hinks further discloses *wherein the locale-sensitive content is placed in a separate file,* (column 3 lines 13-16).

Hinks does not expressly disclose *and the references comprise a pointer to that file.*

However, Sun teaches a reference to a file containing locale-sensitive content (page 97 under "Text Databases (Text Domains)".

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Sun's reference pointer with Hinks' content file. One of

ordinary skill would have been motivated to retrieve text in the local language from a program with placeholders for a localizer to put each string's translation.

As per claim 43, the above rejection of claim 41 is incorporated. Hinks further discloses *wherein the locale-sensitive content is placed in a separate data structure* (column 3 lines 13-16 as cited in claim 5).

Hinks does not expressly disclose references that comprise function calls.

However, Sun teaches *the references comprise function calls that, when executed, obtain the associated locale-sensitive content from the data structure and insert the associated locale-sensitive content into the source code* (page 99 under the header "xgettext and msgfmt Utilities" and page 98 under the header "gettext() and dgettext() Routines").

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Sun's function calls to reference Hinks' data structure. One of ordinary skill would have been motivated to retrieve text in the local language from a program with placeholders for a localizer to put each string's translation.

As per claim 49, Hinks discloses:

A system (FIG. 3) comprising:

compilation means for compiling a computer-servable document written for a particular locale to extract ... characters associated with any original locale-sensitive content (column 3 lines 1-4, and 7-9 as cited in the above rejection of claim 20), *the*

compilation means producing a compiled document with locale-independent elements (FIG. 3 element 377 and column 8 lines 30-34 as cited in the above rejection of claim 20)*; and*

storage means for storing the original locale-sensitive content extracted from the computer-servable document in a data structure separate from the compiled document See column 3 lines 13-16:

> From there, Export/Import (EXPIMP) module parses the resource file into a Translation Table, which is typically stored as a database table.

...further comprising runtime means for populating, at runtime, the compiled document with a desired version of the original locale-sensitive content to reconstruct the computer-servable document, wherein the populating comprises executing ...to obtain the desired version of the associated original locale-sensitive content and to insert the desired version of the associated original locale-sensitive content back into the compiled document. See column 3 lines 50-55:

> Resources for products to be translated are stored in an external resource file in a standard format. Thus, the translation process need not interfere with the executable binary (i.e., main program) of a product. In this fashion, changing a product from one locale to another can be reduced to the simple process of swapping out resource files.

Hinks does not expressly disclose *removing characters* and *wherein the compilation means comprises substitution means for substituting a function in place of associated removed original locale-sensitive content in the compiled document, the function representing the associated removed original locale-sensitive content extracted from the compiled document;*

However, Sun teaches semantic substitution of a string with a function call which represents the original extracted content. See page 99 under the header "xgettext and msgfmt Utilities":

> Once Devguide or a developer has inserted `gettext()` function calls around all user-visible text in an application, `xgettext` can be run on the source files to produce the portable object files.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Sun's function call substitution and population in Hinks' localization system. One of ordinary skill would have been motivated to include nothing specific to one's language and culture in software development and to provide features that facilitate translation of text into other languages (See Sun page 95).

Hinks and Sun do not expressly teach *removing* locale-sensitive content. However, in an analogous environment, Lyapustina teaches that locale-sensitive content can be removed and substituted with a unique identifier string. See column 4 lines 18-22:

> Upon identifying each string, the conversion mechanism generates a unique macro string as a substitute for the original string. The conversion mechanism then substitutes the unique macro string for the identified string in the source code of the computer program.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Lyapustina's unique macro string with Sun's function call. One of ordinary skill would have been motivated to avoid the use of hard coded strings (See Lyapustina column 1 line 64 – column 2 line 7).

As per claim 52, the above rejection of claim 49 is incorporated. Further, Hinks discloses *wherein the locale-sensitive content is translated in to a second version for use in a second locale* (column 3 lines 35-36).

As per claim 55, Hinks discloses:

One or more computer-readable media (column 5 line 44: "main memory") *comprising computer-executable instructions that, when executed, direct a computer to:*

examine source code in a document written for a particular locale (column 3 lines 7-9: "First, the sources are **parsed** by the Export/Import module to a translatable format.");

extract ... characters associated with any original locale-sensitive content from the source code (column 3 lines 1-4: "The Export/Import module itself includes a parsing engine to **extract** strings and translatable information from application programs.");

store the original locale-sensitive content in a separate file (column 3 lines 13-16: "From there, Export/Import (EXPIMP) module parses the resource file into a Translation Table, which is typically stored as a database table.").

All further limitations have been addressed in the above rejection of claim 20.

12. Claims 32-35 and 40 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hinks, Sun, and Lyapustina, and further in view of prior art of record U.S. Patent Application Publication US 2002/0107684 to Gao, filed Feb. 7, 2001 (hereinafter referred to as "Gao").

As per claim 32, Hinks discloses a system (FIG. 3). Hinks further discloses:

at least one computer-servable document stored in a computer-readable medium, the document being written for a particular locale See column 3 lines 7-9:

> First, the **sources** are parsed by the Export/Import module to a translatable format.

Also column 5 lines 59-61:

> Software system 200, which is stored in system memory 102 and on **disk memory** 107, includes a kernel or operating system (OS) 240 and a windows shell 250.

a compiler to automatically extract ... characters associated with any original locale-sensitive content from the document to produce a compiled document containing locale-independent elements See column 3 lines 7-7:

> First, the sources are parsed by the **Export/Import module** to a translatable format.

Also column 3 lines 50-51:

> Resources for products to be translated are stored in an **external resource file** in a standard format.

wherein the compiler stores the original locale-sensitive content in a data structure separate from the compiled document; See column 3 lines 13-16:

> From there, Export/Import (EXPIMP) module parses the resource file into a Translation Table, which is typically stored as a database table.

Hinks also discloses obtaining the associated locale-sensitive content and inserting the associated locale-sensitive content back into the compiled document (column 3 lines 50-55).

Hinks does not expressly disclose *removal* and *substituting a function call in place of associated locale-sensitive content in the compiled document.* All further limitations have been addressed in the above rejection of claim 20.

However, in an analogous environment, Sun teaches *substituting a function call in place of associated ... original locale-sensitive content in the compiled document* (page

99 under the header "xgettext and msgfmt Utilities" as cited in the above rejection of claim 20) ...*wherein the function call is configured such that, when executed at runtime, the function call obtains the desired version of the associated original locale-sensitive content from the data structure and inserts the desired version of the associated original locale-sensitive content back into the compiled document* (page 98 under the header "gettext() and dgettext() Routines" as cited in the above rejection of claim 20).

All further limitations have been addressed in the above rejection of claim 20.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Sun's function call substitution and population in Hinks' localization system. One of ordinary skill would have been motivated to include nothing specific to one's language and culture in software development and to provide features that facilitate translation of text into other languages.

Hinks, Sun, and Lyapustina do not expressly disclose a runtime manager. However, Gao teaches the use of an "Integrated Translation Environment" which populates web documents with locale-sensitive content prior to serving. See page 5 paragraph 0135:

> The Globalisation Content Management component (GCM) controls the overall globalisation workflow, that is the internationalisation followed by a localisation. To globalise a web page, the GCM will first decompose the scripts and content from the page, then start the Internationalisation process with the support of the Internationalisation engine, generate the internationalized templates and resource bundles according to Internationalisation strategies, and manage locales for those resource bundles.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Gao's web localizer with Hinks' internationalization method. One of ordinary skill would have been motivated to present localized content in a document prior

to serving in order to provide meaningful information to a user (see Gao paragraph [0003].

As per claims 33 and 34, the above rejection of claim 32 is incorporated. All further limitations have been addressed in the above rejections of claims 21 and 22, respectively.

As per claim 35, the above rejection of claim 32 is incorporated. Hinks further discloses *wherein the compiler examines source code in the document to determine, from the source code, whether locale-sensitive content is present* (column 3 lines 7-9).

As per claim 40, the above rejection of claim 32 is incorporated. Hinks further discloses a translated version of the locale-sensitive content corresponding to at least one other respective locale (column 3 lines 23-24).

13. Claims 57-60, and 65-72 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hinks, Sun, and Lyapustina as applied to claim 20 above, and further in view of U.S. Patent No. 6,308,212 to Besaw et al. (hereinafter "Besaw").

In regard to claim 57, the above rejection of claim 26 is incorporated. Hinks does not expressly disclose: *receiving a request for the computer-servable document, wherein the request contains an identity of a desired locale, the desired locale associated with the*

desired version of the original locale-sensitive content, wherein the populating comprises, at runtime, executing the function call in the compiled document to obtain the desired version of the associated original locale-sensitive content ***based on the request****; and forwarding the computer-servable document with the desired version of the original locale-sensitive content in reply to the request.* However, Besaw teaches providing a web server with a request containing a locale identity which prompts the system to populate and forward documents in reply to the request. See column 3 line 62 – column 4 line 5. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Besaw's request with Hinks' localization in order to initialize session context information so that it can be shared with other valid applications as suggested by Besaw (see column 1 lines 48-51).

In regard to claim 58, the above rejection of claim 57 is incorporated. Besaw further teaches: *wherein the request is a client request received by a server system, via a network.* See column 3 line 62 – column 4 line 5 as cited above, in combination with Fig. 1.

In regard to claim 59, the above rejection of claim 58 is incorporated. Hinks does not expressly disclose: *wherein the computer-servable document is a web page.* However, Besaw's client is a web browser that displays web pages. See column 3 line 62 – column 4 line 5 as cited above.

In regard to claim 60, the above rejection of claim 58 is incorporated. Besaw further teaches: *wherein the server system implements a multi-layer architecture,* See Fig. 1. *wherein the multi-layer architecture comprises a business logic layer for processing the client request according to an associated problem domain.* See FIG. 3, element 20.

In regard to claims 65-68, the above rejection of claim 52 is incorporated. All further limitations have been addressed in the above rejections of claims 57-60, respectively.

In regard to claims 69-72, the above rejection of claim 55 is incorporated. All further limitations have been addressed in the above rejections of claims 57-60, respectively.

14. Claims 61-64 rejected under 35 U.S.C. 103(a) as being unpatentable over Hinks, Sun, Lyapustina, and Gao as applied to claim 40 above, and further in view of Besaw.

In regard to claims 61-64, the above rejection of claim 40 is incorporated. All further limitations have been addressed in the above rejections of claims 57-60, respectively.

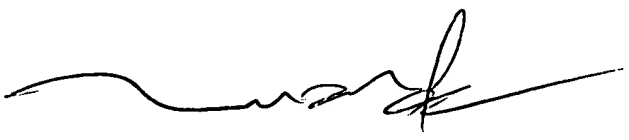
Conclusion

15. Any inquiry concerning this communication or earlier communications from the examiner should be directed to J. Derek Rutten whose telephone number is (571)272-3703. The examiner can normally be reached on M-F 8:30-5.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571)272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

jdr



TUAN DAM
SUPERVISORY PATENT EXAMINER